Future Computing and Informatics Journal

Volume 2
Issue 2 *2017, Volume 2, Issue 2*

Article 10

2017

A robust 3D mesh watermarking algorithm utilizing fuzzy C-Means clustering

Ola M. El Zein
Al-Azhar University, Faculty of Science, Egypt, dr_ola@rocketmail.com

Lamiaa M. El Bakrawy
Al-Azhar University, Faculty of Science, Egypt, dr_lamiaa_el_bakrawy@azhar.edu.eg

Neveen I. Ghali Prof.
Future University, Faculty of Computers and Information, Egypt, Neveen.Ghali@Fue.edu.eg

Available online at www.sciencedirect.com

ScienceDirect

Future Computing and Informatics Journal 2 (2017) 148–156
http://www.journals.elsevier.com/future-computing-and-informatics-journal/

A robust 3D mesh watermarking algorithm utilizing fuzzy C-Means clustering

Ola M. El Zein [a,*], Lamiaa M. El Bakrawy [a], Neveen I. Ghali [b]

[a] *Al-Azhar University, Faculty of Science, Egypt*
[b] *Future University, Faculty of Computers and Information, Egypt*

Received 9 March 2017; revised 17 August 2017; accepted 15 October 2017
Available online 21 November 2017

Abstract

A new robust 3D watermarking algorithm utilizing Fuzzy C-Means (FCM) clustering technique is presented. FCM clusters 3D mesh vertices into suitable and unsuitable choices to insert the watermark without occasioning visible deformation, and also it is tough for the attacker to determine places of the watermark insertion. Two watermarking processes are offered to insert the watermark into 3D mesh models. The first process utilizes topical statistical measurements like average and standard deviation in order to alter the values of vertices to secret watermark data into 3D mesh models, however, the second process utilizes a jumbled insertion planning to insert the watermark inside 3D mesh models utilizing the topical statistical measurements and altering 3D mesh vertices together. Simulation results show that the proposed algorithm is robust. The watermarked 3D mesh models are resistant to several attacks like similarity transforms, noise addition, cropping and mesh smoothing.

Keywords: Fuzzy C-Means; Statistical measures; Local geometry; 3D mesh; 1-Ring neighborhood; RMSE (root mean square error)

1. Introduction

Watermarking is a mechanism utilized to provide the protection for copyright and prove ownership at the domain of data hiding [1]. Generally available digital watermarking mechanisms have concentrated on digital content types such as still images, audio and videos due to spread of these data over internet [2]. The watermarking of 3D models has got less interested than 2D watermarking from researchers since the technology which has been utilized for analysis the video and image cannot be readily appropriated to 3D objects (models) that can be characterized in many ways including NURBS (Non-Uniform Rational Basis Spline), Voxels and 3D meshes [3].

Recently, 3D meshes are greatly utilized in many domains such as virtual reality, computer-aided design (CAD), entertainment means and so on [4]. For that reason, the problem of copyright protection has received more attraction in the domains of research and industry.

The recent algorithms on 3D watermarking are classified to two types: spatial-domain methods and frequency-domain methods [5,6]. The first type inserts the watermark by immediately changing the mesh geometry or connectivity, while the second type inserts the watermark by changing the frequency domain coefficients after mesh transformation [7]. The insertion process in the first type is often easier and faster than in the second type, the inserted watermark is less imperceptible and robust for the operations of 3D meshes. However, the watermarking process in the second type is greater complicated and slower than the first type due to the requirement to transform and reverse transform [6].

According to the extraction operation, watermarking approaches are classified into blind and non-blind extraction [8].

* Corresponding author.
E-mail addresses: dr_ola@rocketmail.com (O.M. El Zein), dr_lamiaa_el_bakrawy@azhar.edu.eg (L.M. El Bakrawy), neveen.ghali@fue.edu.eg (N.I. Ghali).
Peer review under responsibility of the Faculty of Computers and Information, Future University in Egypt

https://doi.org/10.1016/j.fcij.2017.10.007

A blind extraction watermark needs only the secret key in the extraction stage and does not need the original model, but non-blind extraction watermark needs the secret key and original model in the extraction stage [9,10].

This paper focuses on inserting big quantity of hidden data into the original models based on FCM clustering technique without occasioning visible deformation and also it is hard for the attacker to speculate locations of the watermark insertion. This is performed by presenting the algorithm which utilizes the unsupervised mechanism. In this work, FCM is utilized to cluster mesh vertices into proper and improper choices for being watermark carrier based on the feature vector. Then watermark is inserted in the proper vertices that come out from FCM clustering technique based on two methods, the first method utilizes set of topical statistical measurements like average and standard deviation that modifies the vertex value for the vertices of the mesh model to secret watermark data into 3D meshes [10]. The second method for watermark insertion utilizes a jumbled insertion planning which divides watermark into two portions and the first portion is inserted by utilizing the topical statistical measurements [10], whereas the second portion is inserted immediately in the vertices itself [11]. The proposed algorithm is counted development for the main idea presented in Ref. [10] that is achieved by simulation results.

The remainder of this paper is regulated as follows: Briefly reviews to the introductory concepts connecting to the proposed algorithm are introduced in Section 2. In Section 3, the proposed algorithm of 3D mesh watermarking including selection vertices using FCM, insertion and extraction process of watermark is discussed. Experimental results including comparisons with existing techniques are shown in Section 4. Conclusions of this paper are discussed in Section 5.

2. An overview

2.1. Fuzzy C-Means clustering

Clustering is the method of combination a set of data objects into classes or clusters so that objects inside the cluster are like to each other but different from the objects in other clusters [12].

Fuzzy C-Means is considered one of the greatest common fuzzy grouping mechanisms. It was presented by Dunn [13] in 1973 and finally altered in 1981 by Bezdek [14]. FCM is based on minimization of a generalized least-squared errors function. It partitions the objects group V = [v_1; v_2; ...; v_l] into k fuzzy clusters by reducing the total set of squared error of objective function as follows [15,16]:

$$J=\sum_{j=1}^{l}\sum_{i=1}^{k}\left(h_{ij}\right)^{e}d^{2}\left(v_j\,,q_i\right)=\sum_{j=1}^{l}\sum_{i=1}^{k}\left(h_{ij}\right)^{e}\left|\left|v_j-q_i\right|\right|^{2} \qquad (1)$$

where l is the objects group number, k is the clusters number with 2 < k < l, h_{ij} is the membership degree of v_j in the ith cluster, e is the weighting exponent on every membership, q_j is the cluster center of i, $d^2(v_j, q_i)$ is the distance between object v_j and the center of cluster q_i.

The FCM algorithm can be summarized by the following steps:

1: Initialize matrix H = [h_{ij}] with the initial value $H^{(0)}$.
2: At n-step: calculate the cluster center matrix $Q^{(n)} = [q_i]$ with $H^{(n)}$.
3: Update $H^{(n)}$, $H^{(n+1)}$.
4: If $\left\|H^{(n+1)}-H^{(n)}\right\|<\varepsilon$ then end, or go to step 2.

where ε is the threshold of the termination condition.

2.2. 3D mesh model representation

3D mesh model is consisted of a combination of vertices V in Cartesian coordinates and a combination of edges E that linked the vertices. Suppose v_j is the vertex indexed by j and is characterized by its 3D coordinates (X_j; Y_j; Z_j) [17]. Fig. 1 is an example of 3D mesh model.

The group of all vertices that adjacent or neighbors to a vertex V_j is named 1-ring of the vertex. The number of vertices that adjacent to vertex V_j in the 1-ring is the degree of the vertex V_j [18]. Example of vertices that have degree 6 in 1-ring is shown in Fig. 2.

3. The proposed algorithm of 3D mesh watermarking

The watermarking algorithm employs FCM clustering technique to determine the location of vertices to insert the watermark without occasioning perceptual deformation and also makes the watermark be robust against attacks and cannot be detected from the attackers. The proposed algorithm contains three steps as shown in Fig. 3. First step, selection vertices utilizing FCM clustering technique based on the feature vector. FCM is utilized to cluster vertices into appropriate and inappropriate choices for carrier watermark. Second step, insertion process of watermark that inserts a random watermark bits into the selected locations. Third step, extraction process that passes through the same steps of insertion on the receiver side.

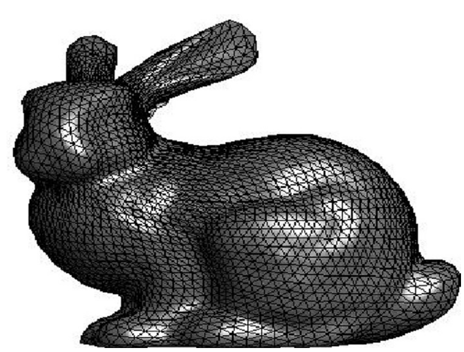

Fig. 1. 3D mesh model.

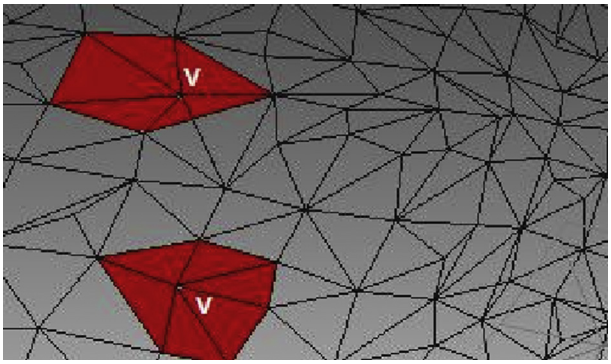

Fig. 2. Example of vertices with degree 6 in 1-ring.

3.1. Selection vertices using FCM

Selection and determination group of vertices as carrier of watermark is considered the first step in the proposed algorithm. It can performed by clustering 3D mesh vertices into proper and improper carrier of watermark utilizing FCM based on the feature vector. Feature vector is a group of angles calculated between the normals and the average normal of the trigonal faces which constitute a 1-ring for a vertex [11]. It is calculated to vertices of degree 6 since most of the vertices in a 3D model have degree 6. Changing or selecting different values of degree will lead to suggest different positions of vertices to insert the watermark that will affect on the performance of the proposed algorithm.

The steps for finding the feature vector for vertices of degree 6 are:

Step 1 Compute normals n_j to all face that is constituted by V and its adjacent vertices as illustrated in Fig. 4.

Step 2 Determine the average n_{avr} of all the previous normals transiting over the vertex V.

$$n_{avr} = \frac{1}{p}\sum_{j=1}^{p} n_j \qquad (2)$$

where, p is the number of its neighbor faces to a vertex V.

Step 3 Compute the angles between the surface normal n_j and average normal n_{avr}, Feature vector F = (θ_1; θ_2; θ_3; θ_4; θ_5; θ_6).

$$\theta_j = \cos^{-1}\left(\frac{n_j \cdot n_{avr}}{|n_j||n_{avr}|}\right) \qquad (3)$$

In order to perform clustering, feature vector is calculated to all vertices of degree 6. It determines the topical geometry of an area. If the area is peak, the measure of angles will be high but in flat areas the measure of angles will be low. Peak and flat areas are ignored to achieve high transparency and consider their vertices as improper watermark carriers. We only take into consideration the vertices of moderate value to be carriers of the watermark, by utilizing FCM to cluster all mesh vertices into three groups according to the values of vertices (low, moderate, and high). The process of calculate feature vector for 3D mesh model is shown in Fig. 5.

3.2. Insertion of watermark

A random sequence of watermark w is inserted in the proper vertices which have moderate value that output from utilizing FCM clustering technique. Two methods are utilized

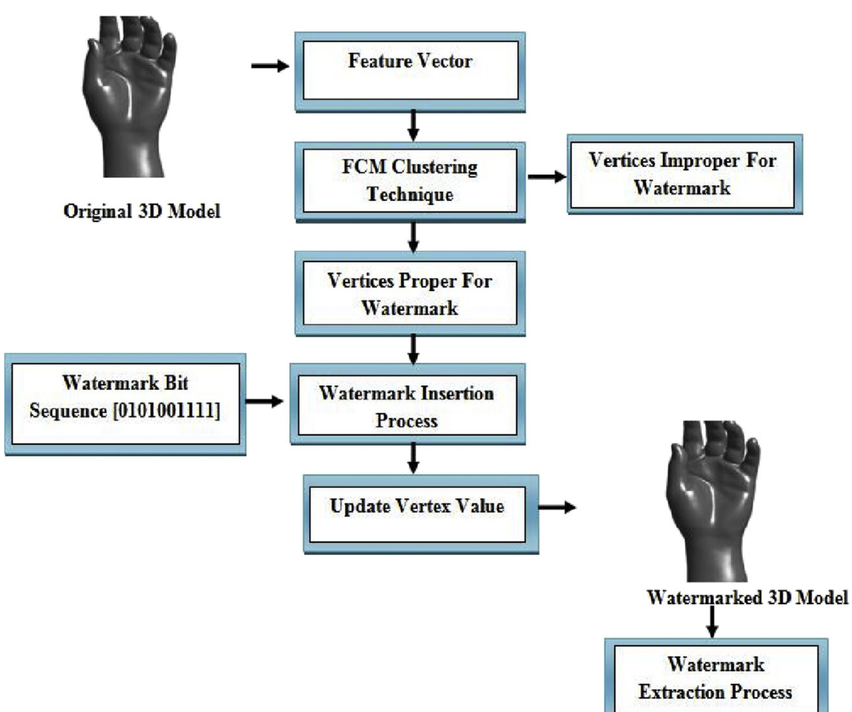

Fig. 3. Diagram of the proposed 3D mesh watermarking algorithm.

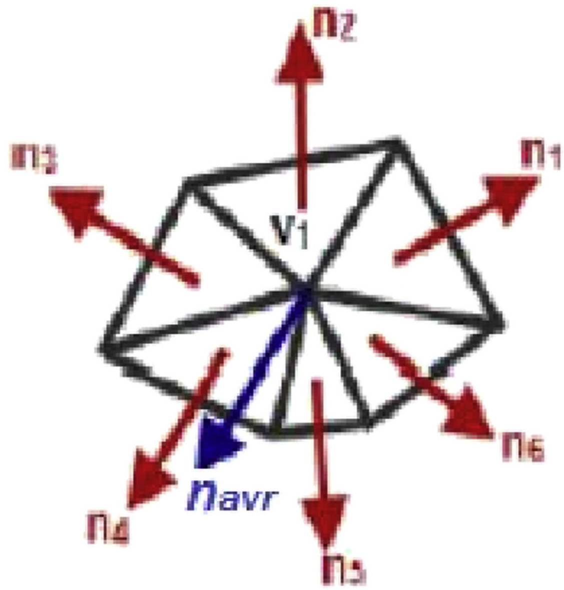

Fig. 4. Description of normals (n_1; n_2; …; n_6) and average normal (n_{avr}) for a 1-ring vertex of degree 6.

to insert the watermark. The first method I inserts the watermark by utilizing the topical statistical measurements. This is performed by searching for the 1-ring $\{NB_i\}$ of proper vertices with degree 6, then calculating their statistical measurements average μ and standard deviation σ. The watermark w_i is inserted by utilizing these two values 0 or 1 according to the following equations:

$$V^*_{x,y,z} = \mu(V_{x,y,z}\{NB_i\}) - 2*\sigma\left(V_{x,y,z}\{NB_i\}\right) \;\; if \;(w_i == 0) \quad (4)$$

$$V^*_{x,y,z} = \mu(V_{x,y,z}\{NB_i\}) + 2*\sigma\left(V_{x,y,z}\{NB_i\}\right) \;\; if \;(w_i == 1) \quad (5)$$

where, $V^*_{x,\,y,\,z}$ is the watermarked vertex for x, y, z coordinates, $V_{x,\,y,\,z}$ is the original vertex for x, y, z coordinates.

The second method II utilizes a novel idea of dividing the watermark into two portions, the first portion (40 binary bits) is inserted by utilizing the topical statistical measurements as shown in the first method while the second portion (24 real numbers) is inserted by modifying the vertices to improve the imperceptibility of the watermark according to the next equation:

$$V^*_{(x,\,y,\,z)} = V_{(x,\,y,\,z)} + a * w \quad (6)$$

where a is the scaling factor (constant value 0.01) and *w* is the watermark. If the length of the watermark changes in the second method for the two portions this will affect the insertion method. Method I and II indicate the two insertion methods.

3.3. Extraction of watermark

First method I does not need the original model in the extraction process, so it is counted as blind watermarking process. The positions of watermark *w* are detected by stratifying the same procedures on the receive side, by computing the 1-ring $\{NB_i\}$ of suitable vertices and calculating their average μ and standard deviation σ of adjacent vertex as shown in the following equation:

$$J_{x,\,y,\,z} = \mu(V^*_{x,\,y,\,z}\{NB_i\}) - 2*\sigma\left(V^*_{x,\,y,\,z}\{NB_i\}\right) \quad (7)$$

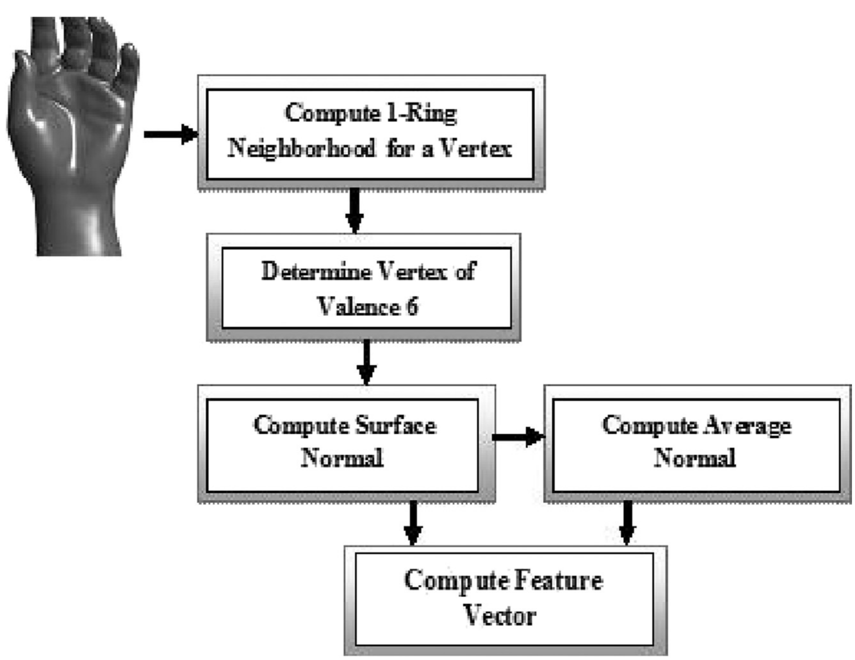

Fig. 5. Process of calculate feature vector for 3D mesh model.

$$K_{x,y,z} = \mu(V^*_{x,y,z}\{N\,B_i\}) + 2{*}\sigma\left(V^*_{x,y,z}\{N\,B_i\}\right) \quad (8)$$

Then compare the suitable vertices value with their current average and standard deviation to extract the watermark bits w. This is done by a changed value to achieve high correlation coefficient against different attacks according to the next equation:

$$if\left(V^*_{x,y,z} \geq J_{x,y,z} - \delta\right) and \left(V^*_{x,y,z} \geq J_{x,y,z} + \delta\right) \;\; (w_i == 0) \quad (9)$$

$$if\left(V^*_{x,y,z} \geq k_{x,y,z} - \delta\right) and \left(V^*_{x,y,z} \geq k_{x,y,z} + \delta\right) \;\; (w_i == 1) \quad (10)$$

Second method II is non-blind watermarking process since the original model is needed in the extraction process. The extraction of watermark for the second method passes over two ways: the first way utilizes the extraction process of method I to extract the first portion of watermark w. The second portion of watermark is extracted by computing the vertex distinction between watermarked and original model as shown in the following equation. The two portions of watermark are merged to supply all watermark series.

$$w = V^*_{x,y,z} - V_{x,y,z}/a \quad (11)$$

4. Experimental results and comparison

The proposed algorithm is executed in Matlab R2014a and has been examined on many 3D mesh models. Five of them are demonstrated in Fig. 6: Bunny (34,835 vertices; 69,666 faces), Hand (36,619 vertices; 72,958 faces), Rabbit (70,658 vertices; 141,312 faces), Venus (100,579 vertices; 201,514 faces) and Dragon (50,000 vertices; 100,000 faces) respectively.

To evaluate the proposed watermarking algorithm, a sequence of experiments are executed to examine the robustness and imperceptibility of watermarking method. The length of watermark that is utilized for insertion is 64 bit series. In this paper, experimental results for method I and method II are compared to Soliman's method [10] and Li's method [2].

4.1. Evaluation of sensitivity

To validate the perceptual goodness of the watermarked models after inserting the watermark. Two measures are utilized: The first measure is Vertex Signal-to-Noise Ratio (VSNR) which determines the perceptual variations between the original and watermarked models. VSNR can be calculated as follows [19]:

$$\mathrm{SNR} = \frac{\sum_{j=1}^{C}\left(X_j^2 + Y_j^2 + Z_j^2\right)}{\sum_{j=1}^{C}\left[\left(X_j - X_j^*\right)^2 + \left(Y - Y_j^*\right)^2 + \left(Z_j - Z_j^*\right)^2\right]} \quad (12)$$

where C is vertices number in the 1-ring neighbor of the vertex, (X_j, Y_j, Z_j) and (X_j^*, Y_j^*, Z_j^*) are the original and altered Cartesian coordinates of the vertices prior and after inserting the watermark.

$$\mathrm{VSNR} = 20\log_{10}(SNR) \quad (13)$$

The other measure is the method proposed by Cignoni et al. in Ref. [20] known as Root Mean Square error (RMSE) which based on a correspondence between each pair of vertices of the models to compare, thus it is limited to the comparison between two meshes sharing the same connectivity. The root mean square error is evaluated as:

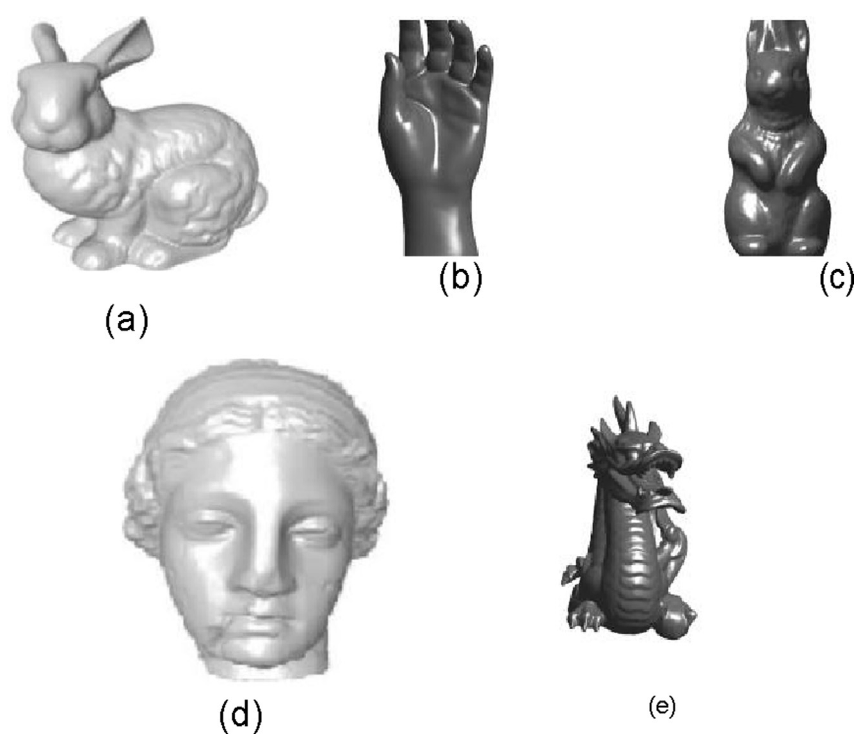

Fig. 6. The original models: (a) Bunny, (b) Hand, (c) Rabbit, (d) Venus, and (e) Dragon.

Table 1
Baseline evaluation of proposed watermarking method.

Mesh models	Method I		Method II		Soliman's method		Li's method
	VSNR	RMSE	VSNR	RMSE	VSNR	VSNR	RMSE
Bunny	119.23	0.31×10^{-3}	122.53	0.25×10^{-3}	106.73	108.40	0.244×10^{-3}
Hand	121.12	0.31×10^{-3}	122.90	0.27×10^{-3}	120.08	120.23	–
Dragon	128.48	0.20×10^{-3}	131.73	0.17×10^{-3}	118.37	116.19	0.306×10^{-3}
Venus	136.02	0.17×10^{-3}	140.16	0.13×10^{-3}	135.53	135.89	0.216×10^{-3}
Rabbit	135.77	0.11×10^{-3}	138.48	0.10×10^{-3}	135.69	133.77	0.261×10^{-3}

$$RMSE = \sqrt{\sum_{j}^{C} \left\| V_j - V_j^* \right\|^2} \tag{14}$$

The values of VSNR and RMSE for the examined models after inserting the watermark in the vertices that come out from utilizing FCM clustering technique are shown in Table 1. It shows that the proposed method II gives the greatest value of VSNR and the lowest value of RMSE compared to proposed method I and the work proposed by Refs. [10] and [2]. This means method II preserves better visual quality than method I, Soliman's method [10] and Li's method [2]. Fig. 7 shows the watermarked 3D mesh models after inserting 64 binary bits watermark by the proposed method I.

4.2. Evaluation of robustness

Watermarking algorithm should be able to resistant the malicious or non-malicious attacks. Various attacks like similarity transforms, random noise addition, mesh smoothing and cropping operations are applied on the watermarked 3D mesh models to indicate the robustness of watermarking algorithm.

- Similarity transformations: It contains uniform scaling, translation, rotation, and collections of these three processes. The proposed algorithm is completely resistance to uniform scaling, translation and rotation and gives correlation coefficient of 1.
- Random noise addition: adding random noise to the watermarked 3D mesh model with various noise level (sigma) varies from (S = 0.1%, 0.3%, 0.5%, 0.9%).
- Mesh smoothing: Laplacian smoothing is performed on the watermarked 3D mesh models with (λ = 0.1 and various iteration Itt = 5, 10, 20, 30).
- Cropping attack: cutting part from 3D mesh model with various level specifying the proportion of cropped vertices (C = 10%, 50%, 70%).

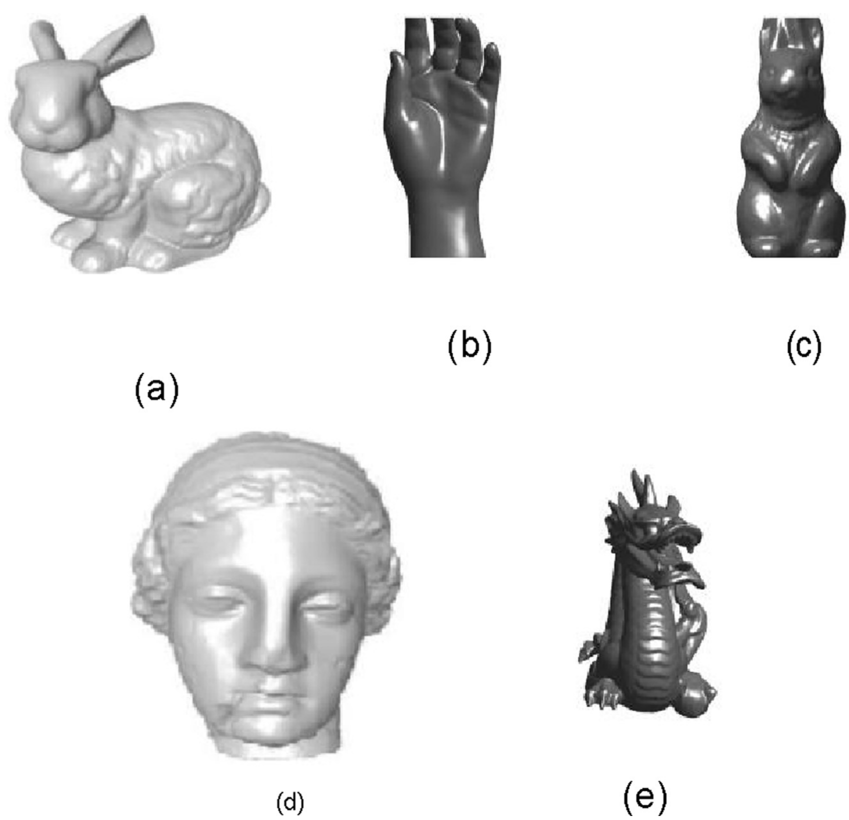

Fig. 7. The Watermarked models: (a) Bunny, (b) Hand, (c) Rabbit, (d) Venus, and (e) Dragon.

Different distortion attacks are performed on dragon model as shown in Fig. 8.

In this work, the correlation coefficient [21] is utilized to estimate the robustness of the proposed algorithm. It is utilized to measure the likeness between the extracted and the original watermark. The correlation coefficient CN value is a number between −1 and +1 and is evaluated by the next equation.

$$CN = \frac{\sum_{j=1}^{J-1}\left(w_j^* - \overline{w^*}\right)\left(w_{j-\overline{w}}\right)}{\sqrt{\sum_{j=1}^{J-1}\left(w_j^* - \overline{w^*}\right)^2 \sum_{i=1}^{J-1}\left(w_{j-\overline{w}}\right)^2}} \tag{15}$$

where J is the number of watermark, $\overline{w^*}$ and $\overline{w}$ indicate respectively the averages of the extracted watermark w_j^* and the original inserted watermark w_j Tables 2–4 illustrate the correlation coefficient outcomes of the two proposed methods for many kinds of attack with different parameters compared with Soliman's method and Li's method [2]. It also shows the impact of noise on watermarked model by computing RMSE at various noise parameters.

Random noise is applied to the watermarked model five times using different noise level and report the median as shown in Table 2. Both two proposed methods are fairly resistant to the noise attacks under a noise level of 0.9 more than Soliman's method. In addition, method I is more robust than method II, Soliman's method and Li's method. Table 2 shows that two proposed methods are superior than Soliman's method and Li's method in resisting noise attacks in most models. It means that the two proposed methods achieve high robustness.

Table 3 illustrates the rendering of the watermarking methods after applying the smoothing attacks. The two proposed methods achieve high results in resisting smoothing

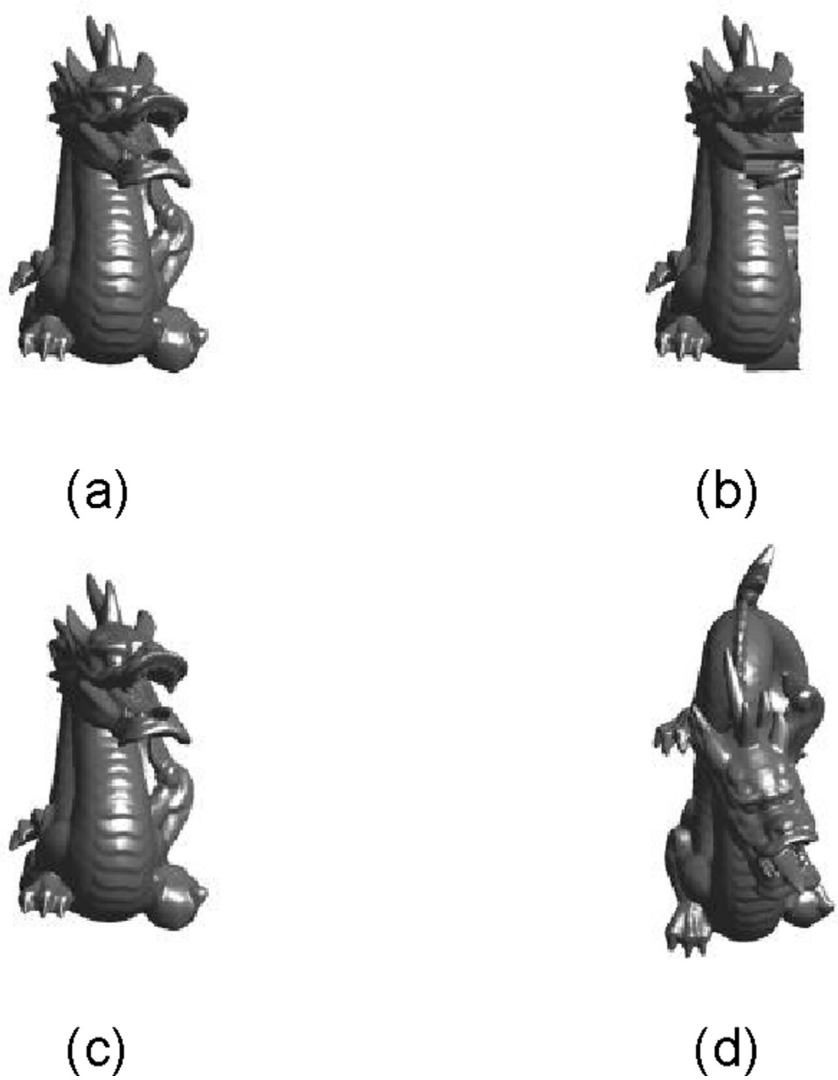

Fig. 8. Dragon model watermarked by method I and attacked by: (a) adding noise with 0.1%, (b) cropping with 10%, (c) smoothing with Itt 20, and (d) similarity transforms.

Table 2
Robustness against additive noise attacks.

Models	S	Method I		Method II		Soliman's method		Li's method
		RMSE	CN	RMSE	CN	CN	CN	CN
Bunny	0.1%	0.65×10^{-3}	1	0.6365×10^{-3}	0.998	0.774	0.871	1
	0.3%	0.18×10^{-2}	1	0.17×10^{-2}	0.978	0.770	0.860	0.90
	0.5%	0.29×10^{-2}	0.968	0.29×10^{-2}	0.887	0.653	0.771	0.68
	0.9%	0.52×10^{-2}	0.581	0.52×10^{-2}	0.554	0.506	0.544	–
Hand	0.1%	0.65×10^{-3}	1	0.63×10^{-3}	0.996	0.864	0.886	
	0.3%	0.18×10^{-2}	1	0.18×10^{-2}	0.987	0.859	0.859	
	0.5%	0.29×10^{-2}	0.967	0.29×10^{-2}	0.906	0.688	0.688	
	0.9%	0.52×10^{-2}	0.617	0.52×10^{-2}	0.601	0.432	0.426	
Dragon	0.1%	0.61×10^{-3}	1	0.60×10^{-3}	0.998	0.949	0.893	1
	0.3%	0.17×10^{-2}	1	0.17×10^{-2}	0.983	0.927	0.882	1
	0.5%	0.29×10^{-2}	0.938	0.29×10^{-2}	0.912	0.726	0.715	0.75
	0.9%	0.52×10^{-2}	0.805	0.52×10^{-2}	0.617	0.449	0.448	–
Venus	0.1%	0.60×10^{-3}	1	0.59×10^{-3}	0.999	0.964	0.927	1
	0.3%	0.17×10^{-2}	1	0.17×10^{-2}	0.978	0.941	0.923	0.87
	0.5%	0.29×10^{-2}	0.935	0.29×10^{-2}	0.920	0.715	0.701	0.87
	0.9%	0.52×10^{-2}	0.753	0.52×10^{-2}	0.614	0.441	0.452	–
Rabbit	0.1%	0.58×10^{-3}	1	0.58×10^{-3}	0.995	0.927	0.909	1
	0.3%	0.17×10^{-2}	0.968	0.17×10^{-2}	0.955	0.893	0.830	0.72
	0.5%	0.29×10^{-2}	0.909	0.29×10^{-2}	0.871	0.646	0.608	0.25
	0.9%	0.52×10^{-2}	0.473	0.52×10^{-2}	0.488	0.380	0.374	–

Table 3
Robustness against smoothing attacks ($\lambda = 0.1$).

Models	Itt	Method I		Method II		Soliman's method		Li's method
		RMSE	CN	RMSE	CN	CN	CN	CN
Bunny	5	0.44×10^{-3}	1	0.42×10^{-3}	0.996	0.968	0.968	–
	10	0.65×10^{-3}	1	0.64×10^{-3}	0.991	0.938	0.908	1
	20	0.11×10^{-2}	0.870	0.11×10^{-2}	0.833	0.767	0.795	–
	30	0.15×10^{-2}	0.870	0.15×10^{-2}	0.824	0.415	0.581	0.77
Hand	5	0.34×10^{-3}	1	0.33×10^{-3}	0.997	0.939	0.882	
	10	0.47×10^{-3}	1	0.46×10^{-3}	0.993	0.939	0.882	
	20	0.76×10^{-3}	0.817	0.76×10^{-3}	0.814	0.906	0.813	
	30	0.99×10^{-3}	0.781	0.99×10^{-3}	0.771	0.547	0.397	
Dragon	5	0.64×10^{-3}	1	0.63×10^{-3}	0.992	0.910	0.882	–
	10	0.10×10^{-2}	0.968	0.10×10^{-2}	0.981	0.843	0.847	1
	20	0.18×10^{-2}	0.968	0.18×10^{-2}	0.959	0.596	0.561	–
	30	0.24×10^{-2}	0.845	0.24×10^{-2}	0.852	0.402	0.461	0.80
Venus	5	0.45×10^{-3}	0.935	0.36×10^{-3}	0.996	0.910	0.910	–
	10	0.58×10^{-3}	0.935	0.58×10^{-3}	0.989	0.910	0.910	1
	20	0.99×10^{-3}	0.870	0.98×10^{-3}	0.976	0.819	0.882	–
	30	0.13×10^{-2}	0.870	0.13×10^{-2}	0.932	0.591	0.792	0.94
Rabbit	5	0.10×10^{-3}	1	0.99×10^{-4}	0.996	0.728	0.728	–
	10	0.13×10^{-3}	1	0.13×10^{-3}	0.992	0.728	0.705	1
	20	0.24×10^{-3}	1	0.24×10^{-3}	0.983	0.705	0.658	–
	30	0.35×10^{-3}	1	0.34×10^{-3}	0.977	0.547	0.547	1

attacks at different iterations compared with the Soliman's method and Li's method. In addition, method I is more robust than method II, Li's and Soliman's methods in most models.

For evaluating the robustness against cropping attacks, various cropping levels are performed on watermarked 3D mesh models. Table 4 shows the robustness against cropping attacks. Both two proposed methods and soliman's method are robust at 10% cropping level in most models, but the robustness of method II and soliman's method are less than the robustness of method I by increasing the cropping level in most models. The results of RMSE that listed in Tables 2−4 indicate that the two proposed methods maintain the quality of the image.

Table 4
Robustness against cropping attacks (C%).

Models	C%	Method I	Method II	Soliman's method	
		CN	CN	CN	CN
Bunny	10%	0.936	1	0.851	0.908
	50%	0.599	0.927	0.660	0.795
	70%	0.384	0.778	0.527	0.607
Hand	10%	0.870	0.814	0.969	0.882
	50%	0.653	0.239	0.969	0.882
	70%	0.417	0.028	0.385	0.322
Dragon	10%	0.870	0.813	0.700	0.757
	50%	0.553	0.218	0.353	0.197
	70%	0.553	0.218	0.173	−0.0173
Venus	10%	0.845	0.823	0.686	0.589
	50%	0.434	0.236	0.016	0.016
	70%	0.255	0.199	0.016	−0.2019
Rabbit	10%	0.871	1	0.659	0.676
	50%	0.626	0.277	0.472	0.370
	70%	0.525	0.038	0.243	0.353

5. Conclusions

In this work, a new 3D watermarking algorithm utilizing FCM is proposed. 3D mesh models are watermarked by altering the vertices chosen by FCM as proper watermark carrier without occasioning perceptible distortion. Two suggested methods are proposed to insert the watermark into the 3D mesh models founded on proper vertex chosen to enhance their sensitivity and robustness. The first suggested method I achieves higher results than the second method in terms of robustness and also it maintains reasonable results of sensitivity while the second suggested method II achieves higher results than the first method in terms of sensitivity. Experimental outcomes shown that the suggested algorithm achieved the demands of watermarking like sensitivity and robustness.

References

[1] Soliman MM, Hassanien AE, Onsi HM. An adaptive watermark-ing approach based on weighted quantum particle swarm optimization. Neural Comput Appl (Springer) 2016;27(2):469−81.
[2] Li S, Ni RR, Zhao Y. A 3D mesh watermarking based on improved vertex grouping and piecewise mapping function. J Inf Hiding Multimedia Signal Process January 2017;8(1):97−108.
[3] Abdallah EE, Hamza AB, Bhattacharya P. Watermarking 3D models using spectral mesh compression. Signal, Image Video Process 2009;3(4):375−89.
[4] Abderrahim Z, Techini E, Bouhlel MS. Progressive compression of 3D objects with an adaptive quantization. (IJCSI) Int J Comput Sci Issues March 2013;10(1):504−11.
[5] Hachani M, Ouled zaid A, Puech W. Feature-based image watermarking algorithm using SVD and APBT for copyright protection. Future Internet 2017;9(13):l−15.
[6] Soliman MM, Hassanien AE, Onsi HM. A robust 3D mesh watermarking approach using genetic algorithms. Springer Int Publ Cham, Article Adv Intelligent Syst Comput 2015;323:731−41.
[7] Zhang Y, Wang C, Wang X, Wang M. Robust mesh data hiding based on irregular wavelet transform. Eur Signal Process Conf 2013;12(1):l−5.
[8] Surya Prakasa Rao R, Rajesh Kumar P. PSO based lossless and robust image watermarking using integer wavelet transform. Glob J Comput Sci Technol (F) Graph Vis 2017;17(1):1−13.
[9] Feng X, Liu Y, Fang L. Digital watermark of 3D CAD product model. Int J Secur Its Appl 2015;9(9):305−20.
[10] Soliman MM, Hassanien AE, Onsi HM. A blind 3D watermarking approach for 3D Mesh using Clustering Based Methods. Int J Comput Vis Image Process April−June 2013;3(2):43−53.
[11] Motwani RC, Motwani MC, Harris FC, Bryant BD. Watermark embedded optimization for 3D mesh objects using classification based approach. Int Conf Signal Acquis Process 2010:128−9. https://doi.org/10.1109/ICSAP.2010.83.
[12] Madhulatha TS. Improved K-mean clustering algorithm for prediction analysis using classification technique in data mining. Int J Comput Appl January 2017;157(6):35−40.
[13] Dunn JC. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. J Cybernet 1973;3:32−57.
[14] Bezdek JC. Pattern recognition with fuzzy objective function algorithms. MA, USA: Kluwer Academic Publishers Norwell; 1981.
[15] Lu Y, Ma T, Yin C, Xie X, Tian W, Zhong Z. Implementation of the fuzzy C-Means clustering algorithm in meteorological data. Int J Database Theory Appl 2013;6(6):1−18.
[16] Yang Y, Huang S. Image segmentation by fuzzy C-Means clustering Algorithm with A Novel penalty term. Comput Inf 2007;26:17−31.
[17] Zafeiriou S, Tefas A, Pitas I. Blind robust watermarking schemes for copyright protection of 3D mesh objects. IEEE Trans Vis Comput Graph 2005;11(5):596−607.
[18] Motwani MC. Third generation 3D watermarking: applied computational intelligence techniques. In: A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science and Engineering; July 2011.
[19] Soliman MM, Hassanien AE, Onsi HM. Robust watermark-ing approach for 3D triangular mesh using self organization map. In: Submitted in federated confrence on computer science and information system, krakw, Poland, pp. 99−104; September 2013.
[20] Cignoni P, Rocchini C, Scopigno R. Metro: measuring error on simplified surfaces. Comput Graph Forum 1998;17(2):167−74.
[21] Wanga K, Lavouea G, Denisb F, Baskurta A. Robust and blind mesh watermarking based on volume moments. Comput Graph 2011;35(1):1−19.